

Implementing a service, particularly a financial service, in a network involving mobile terminals

5 The invention concerns the technical field of using mobile terminals for accessing certain services. Especially the invention concerns the task of facilitating easy and flexible use of financial services through the mobile terminals of a network.

10 Mobile terminals first gained truly widespread acceptance in the form of mobile telephones, which has had a strong influence on the development of user interfaces for mobile terminals. The most widespread format of a user interface in a mobile terminal consists of a microphone, a loudspeaker, an alarm buzzer, a relatively small display and a keypad with a slightly more than a dozen individual keys.

15 Attempts to provide versatility to the selection of services that a user can access through his mobile terminal have repeatedly encountered difficulties that arise from the limited size and limited functionalities of the traditional user interface. Exemplary solutions of this kind are presented e.g. in patent publications EP 0 972 275 and WO 96/13814. As an illustrative example we may consider a user that wants to check the balance of his bank account by using text messages, like the well-known SMS messages (Short Message Service). First the user must invoke the SMS application program of his mobile terminal and select the functionality of writing an SMS message. Then he must use the small keypad of his mobile terminal to type in an inquiry message that includes at least a message type identifier that identifies the message as a balance check, the number of the account the balance of which is to be checked, a username and a password. After having completed the message, the user must select a "transmit message" functionality, type in the SMS service number of his bank or scroll through a list or previously stored numbers to find it there, and release the message for transmission.

20 Typically the number of key presses that are required for a complete balance check operation is in the order of several dozens. As an additional drawback the process includes several inherent error sources: for example a single erroneous key press during typing of the message will cause the whole operation to fail. If the SMS-based service employs usernames and passwords for providing security, it requires the user to remember these or to carry around a paper slip or similar memory aid. It is possible to leave out usernames and passwords if the service relies only on the

capability of a cellular radio system in identifying subscribers, but such a service is only as secure as the cellular radio system, which frequently is not enough.

It is an objective of the present invention to provide a method and apparatus for offering easy-to-use, secure and versatile services to the users of mobile terminals.

5 An additional objective of the invention is that each user can tailor the services so offered to suit his personal needs without sacrificing the ease of use. A further objective of the invention is that the services can be offered to users substantially irrespective of the types of mobile terminals they are using. A yet further objective of the invention is that the services so offered could be open for modification by
10 both users and service providers, according to the needs that concerns every particular service.

The objectives of the invention are achieved by allowing the mobile terminals of users to be programmed so that only a single key press, a very short sequence of key presses or a similar very simple activation of the user interface launches the use of a
15 service in a way which the user and/or the service provider has configured beforehand.

A method according to the invention is characterized by the features that are recited in the characterizing part of the appended independent claim directed to a method.

The invention applies also to a mobile terminal. A mobile terminal according to the
20 invention is characterized by the features that are recited in the characterizing part of the appended independent claim directed to a mobile terminal.

Additionally the invention applies to a system that comprises mobile terminals and at least one server for providing services to the users of the mobile terminals. A system according to the invention is characterized by the features that are recited in
25 the characterizing part of the appended independent claim directed to a system.

Various advantageous embodiments of the invention are described in the depending claims.

The present invention is based on the insight that despite of a veritable proliferation of services that are available for the users of mobile terminals, each user will have a
30 relatively small selection of services that he will use regularly. It is therefore possible to associate such often used services with certain parts or features of even a relatively simple user interface, so that a minimal interaction by the user, such as pressing one key, executing a simple sequence of key presses, or tapping one icon,

can be unambiguously associated with the user's intention to use a particular service. When the mobile terminal notices that the user gives such a dedicated yet simple activation command, it launches the use of the appropriate service, executing independently even complicated sequences of operations so that as a result the user
5 gets the service he wanted.

In the context of the present invention we assume that the services involve not only the mobile terminal itself but a system that comprises a cellular radio network, potentially a wired data transmission network as well as fixed installations such as service providers' servers. Typically using a service means that there is some
10 communication between a mobile terminal and a server through the cellular radio network and the potential wired data transmission network. As an extreme case at least certain occasions of using a certain service may include actions of a mobile terminal only, but even in such cases the mobile terminal typically utilizes information that it has obtained earlier from a server provider's server. Another
15 extreme case is such where the mobile terminal has only the function of a command interpreter and link, so that an activation of a service causes the mobile terminal to transmit a certain command message, and all other activities related to the use of the service are performed at the server.

A central feature of the present invention is the ability of a user and/or a service
20 provider to tailor a certain service exactly to the needs of an individual user, without compromising the ease of use that follows from the simple way of activating the service at the user's mobile terminal. This aspect of the invention is called the service definition aspect. It encompasses a multitude of variations of how should the actual task of defining and tailoring the service be accomplished. Three basic
25 branches of such variations are dynamic definition, programmatic definition and definition by the service provider.

Considering financial services according to the invention, four main categories of usage can be defined. Informational usage refers to services where the user obtains some useful simple information, like the current balance of a certain account.
30 Analytical usage refers to services where the user obtains more descriptive information about the status and/or history of his assets. Advisory usage refers to services through which the user can obtain suggestions about what should he do with his assets. Transactional usage refers to services through which the user can commit financial transactions.

Device independence is achieved in the invention through the use of a specific execution language for defining the operations that various devices must accomplish when using the services in question. The only thing that is then required from the various devices is that they are either capable of understanding and
5 executing said specific execution language, or that they rely on other devices that convert instructions given in said specific execution language into some other, more standardized form of communication.

The exemplary embodiments of the invention presented in this patent application are not to be interpreted to pose limitations to the applicability of the appended
10 claims. The verb "to comprise" is used in this patent application as an open limitation that does not exclude the existence of also unrecited features. The features recited in depending claims are mutually freely combinable unless otherwise explicitly stated.

The novel features which are considered as characteristic of the invention are set
15 forth in particular in the appended claims. The invention itself, however, both as to its construction and its method of operation, together with additional objects and advantages thereof, will be best understood from the following description of specific embodiments when read in connection with the accompanying drawings.

- Fig. 1 illustrates a framework for application of the present invention,
- 20 fig. 2 illustrates schematically a mobile terminal according to an embodiment of the invention,
- fig. 3 illustrates the roles of execution language and its handling,
- fig. 4 illustrates an example of dynamically defining a service,
- fig. 5 illustrates programmatic service definition with a text editor,
- 25 fig. 6 illustrates programmatic service definition with an application development system with a graphical user interface,
- fig. 7 illustrates a mobile terminal based service execution scenario,
- fig. 8 illustrates a distributed service execution scenario in a mobile terminal and server,
- 30 fig. 9 illustrates a modification of a distributed service execution scenario in a mobile terminal and server,
- fig. 10 illustrates communication between several entities during the use of a financial service according to the invention,
- fig. 11 illustrates the concept of branching from a script to client programs and
- 35 fig. 12 illustrates certain system level aspects of the invention.

Fig. 1 illustrates the framework within which services are typically used in accordance with the invention. A user 101 has at his disposal a mobile terminal 102, which is capable of communicating wirelessly with a cellular radio network 103. From the last-mentioned there is a bidirectional data transfer connection to a wired data network 104, to which also a service provider's server 105 is connected. An operator's interface 106 is available for controlling the operation of the server 105.

The most typical case of using a service according to the invention proceeds as follows. At step 111 the user 101 gives a service activation signal through a user interface of the mobile terminal 102. This causes the mobile terminal 102 to execute a series of operations, the result of which is a certain message that the mobile terminal 102 transmits to the cellular radio network 103 at step 112. At step 113 the cellular radio network 103 forwards the message to the wired data network 104, which forwards it further to the server 105 at step 114. The server 105 receives the message from the mobile terminal 102 and responds by performing certain operations, which typically results into a response message being sent from the server 105 to the mobile terminal 102 at step 121. Steps 122 and 123 describe the transmission of the response message through the wired data network 104 and the cellular radio network 103 to the mobile terminal 102. At step 124 the mobile terminal 102 executes certain final operations that include at least displaying to the user 101 certain information that came in the response message.

The present invention aims at arranging the operation of all parts of the system shown in Fig. 1 so that step 111 would be as simple as possible, comprising only minimal required interaction from the user 101. This must be accomplished together with the equally important goal according to which at step 124 the user gets exactly the kind of personalized response that he wanted to get by using this particular service.

Fig. 2 is a schematic representation of certain parts of a mobile terminal that have importance to using services in accordance with the present invention. Each mobile terminal has a processor 201 that executes programs stored in a program memory 202. The processor 201 also communicates with the user interface components 203 of the mobile terminal, e.g. for receiving key commands through a keypad and for displaying information on a display screen, through certain user interface drivers 204. Additionally the mobile terminal comprises a radio transceiver 205 for implementing wireless communications with a cellular radio network.

In a typical prior art mobile terminal the relationship between certain input (key commands) that the processor 201 received through a user interface and a corresponding function was tightly fixed, with flexibility existing only in the form of certain softkeys. With a fixed relationship we mean that e.g. a key press on an alphanumeric key "3def" always caused the processor 201 to receive one of the characters "3", "d", "e" or "f" depending on the number of consecutive presses on the same key, but nothing else. The known softkeys have been keys next to a display unit, so that the processor has been able to indicate, with a word or symbol appearing on the display, the current effect of the key. However, the "programmable" different effects of softkeys have still been very strictly fixed in the factory-time programming of the mobile terminal.

According to the present invention a personalized service must be available for use, i.e. it must be possible to make the mobile terminal execute a certain customized series of actions, by only pressing one key, by executing a very simple sequence of key presses or by otherwise performing a very simple activating action through the user interface. Therefore the invention requires the program memory 202 (or the UI drivers section 204) to comprise an easily reprogrammable UI command interpretation unit 211. Associating a service with a certain simple activating action through the user interface means that the UI command interpretation unit 211 is programmed to respond to such a simple activating action by issuing a command to the processor 201 to start the customized series of actions that constitutes the mobile terminal's part of such a service.

Blocks 212 and 213 shown in fig. 2 are related to the concept of an execution language. According to the invention, it is advantageous to specify a certain execution language that constitutes a standardized, device-independent way of describing actions to be taken by a device that takes part in providing a service. The execution language should allow developers to define the creation and execution of platform-independent and parameterized processes in a uniform way and provide the means to hook these processes, or activities, with each other into entities that together form complete instructions for the operation of the devices involved. The execution language must comprise at least the suitable language means for defining the business logic of single, independent activities. Additionally it is advantageous if the execution language provides means for configuring activities in various ways: for example limits may be placed on time of execution of an activity, and an activity may be characterised by its requirements concerning execution order, like whether

an activity may be run concurrently or in parallel with other activities or whether sequential execution is required.

5 Mechanisms defined in the execution language should include exception signalling, as well as support for receiving input and producing output in an application-independent way. The exact form and content of the execution language is not important to the present invention, as long as the execution language serves to achieve the described objectives.

10 Fig. 3 illustrates further the role of an execution language in defining a service. We assume that there exists a certain way of defining a service, i.e. generating a definition of what should certain device(s) do in order to provide a user with the service he wants. Some alternative ways of defining the service will be described later; for the time being it suffices to assume that a certain service-defining process exists. This process has been schematically represented as 301 in fig. 3.

15 The result of the service-defining process 301 is an execution language script 302. Taking into account the assumption of device independency, the execution language script 302 is as such not directly suitable for execution by a processor. On the other hand the execution language script 302 is readily suitable for storing in a memory and for communicating from one device to another according to some suitable communication protocol. The execution language script 302 is a kind of
20 programmatic representation of the service in a nice, compact packet.

In order to convert an execution language script 302 into a set of executable instructions for a processor to execute, a process of execution language script parsing 303 is needed. The parsing step 303 produces an executable program 304. The program 304 does not need to be self-contained and fully processable in the
25 sense that it would always consist of only processor-executable instructions. For example certain passages of execution language may remain encapsulated at certain parts of the program 304, for example if they represent complicated features that will only be needed under very rarely occurring circumstances so that it is more economical to parse them only if they are really needed, or if they contain passages
30 of execution language that must be transmitted to another device or process to be parsed and executed there. The executable program is then passed on to an execution process proper 305, which causes the device(s) in question to act so that the user gets the service he wanted.

Referring back to fig. 2, the program memory 202 of the mobile terminal is shown to comprise certain memory space 212 reserved for storing execution language scripts. Additionally the program memory 202 of the mobile terminal is shown to store an execution language parser program, through the execution of which the processor 201 can parse execution language scripts into executable programs.

A further functionality that a mobile terminal according to the invention will need is a communication method, because acting according to the user's wishes for providing a service will inevitably include exchanging information with other devices, particularly with servers. The messaging application 214 stored in the program memory 202 contains instructions through the execution of which the processor 201 is capable of transmitting and receiving messages that are related to the services that the user wishes to use. Typically certain points in an executable program that was a result of parsing an execution language script trigger the processor to activate the messaging functionality and to use it for communication with other devices.

At the priority date of this patent application the most widespread and most readily available method for communicating small pieces of data to and from mobile terminals is SMS (Short Messaging Service) messages. For the purposes of the present invention the selection of a communication method is not important: it may well be SMS, but other kinds of messaging applications like MMS (Multimedia Message Service) could as well be used. Because the specific features of a messaging application have little practical significance to their use together with the present invention, it is easy and straightforward to adapt the present invention to utilize also any future messaging applications that are yet unknown.

Next we will consider certain more detailed ways of implementing the service definition process that is schematically represented as 301 in fig. 3. There are at least three mutually alternative approaches to service definition: dynamic definition, programmatic definition and definition by the service provider.

Dynamic definition in general means that the user instructs a device to follow and memorize his actions when he performs manually an act of using a service. Fig. 4 is a schematic representation of a simple exemplary series of events during dynamic definition of a service. In this case the user wants to define a service for inquiring for a piece of information, like an account balance. At step 401 the user initiates dynamic service definition, which causes the normal, manually operated functions of the terminal device to call a dynamic service definition application at step 402

and to put is into a state of readiness. After the dynamic service definition application has been invoked at step 402, it monitors the actions of the user and the manually operated routines in order to find out, how should the dynamically defined service proceed.

- 5 At step 403 the user initiates a message inputting application. When this action of the user is forwarded to the dynamic definition application at step 404, it stores at step 405 an execution language representation of a step of initiating automatic generation of a message. At step 406 the user types in the contents of a query message, which when forwarded at step 407 causes the dynamic definition
10 application to store the contents of the query at step 408. At step 409 the user decides that the message is complete and initiates message transmitting; again information about this action is forwarded to the dynamic definition application at step 410 and it stores at step 411 an execution language representation of a step of initiating message transmitting.
- 15 Steps 412, 413 and 414 represent selection of a number into which the query message is to be transmitted and storing a corresponding number selection command in the dynamically defined service. Steps 415, 416 and 417 represent the user giving his final permission to transmitting the query message, which permission is stored in the dynamically defined service as a command to transmit
20 the query message to the stored number.

- Shortly after the query message has been sent, the mobile terminal receives a response from the server to which it sent the query. At step 418 the terminal device informs the user about a received response. Information about a received response is also given to the dynamic definition application at step 419, which causes it to store
25 at step 420 an execution language representation of the fact that a response message is to be expected. At step 421 the user commands the terminal to display the response message, which command when forwarded to the dynamic definition application at step 422 causes it to store an automatic initiation of response displaying at step 423. After the response has been displayed to the user at step 424,
30 the user gives at step 425 a command to end the dynamic definition of a service. This command is forwarded at step 426 to the dynamic definition application, which stores the completed script at step 427.

- After an execution language script has been stored, it is necessary to associate it with a certain simple activation command through which the user wishes to activate
35 the use of the service at later occasions. Typically after the steps shown in fig. 4 the

mobile terminal will prompt the user to press the key that he wishes to associate with the newly generated script, or otherwise give an example of the activation command that should be used. It is also possible that the stored script automatically appears as a representative icon in a graphical user interface, so that activating that icon will trigger the use of the corresponding service. The dynamic definition application then stores information about the given activation command, and reprograms the programmable user interface command interpreter (211 in fig. 2) so that when that particular command is received through the interface, a the processor of the mobile terminal is instructed to initiate a process of parsing the execution language script and executing the resulting executable program.

Certain considerations must be made regarding the "event listener" or dynamic definition application that is responsible for dynamically converting certain actions into passages of execution language script. The dynamic definition application must have a quite low level access to what is happening in the mobile terminal: for example in the case of fig. 4 it should be able to detect single key presses and the occurrence of a response message having been received. This tends to imply that the dynamic definition application can hardly be completely device independent. If it were only a feature of a certain device independent high level application program, it could well be used for tracking events that concern directly the activities of such an application program, but it could not have access to what is going on outside the specific application program.

Another consideration about the dynamic definition application is its dependency or non-dependency on the strict validity of certain "navigation paths", which term refers to the actual physical actions that a user made during the dynamic definition of a service. Physically speaking, when the user e.g. initiates the message inputting application, he typically presses one key that may well be a softkey so that the action it causes depends on the context; at that moment the effect of the softkey was just indicated to be the initiating of a message inputting application. When accomplishing its event tracking task, the dynamic definition application should not store just the information "softkey XX was pressed", because later when the dynamically defined service is used, the context may be different so that a service that just emulated the press of a softkey at a certain moment might well cause something unexpected to happen. The dynamic definition application should be smart enough to notice that by pressing that softkey the user meant to initiate the message inputting application, so that in the resulting execution language script it

includes a command "initiate message generation" instead of "emulate pressing softkey XX".

Figs. 5 and 6 illustrate schematically two exemplary possibilities of programmatic service definition. It is possible to use a conventional text editor 501 to write a service description as if it was a conventional computer program. Depending on the definitions of the execution language it may be possible to write a service description directly in the execution language, but it is likewise possible to write the service description in some more common and more easily comprehensible programming language and use a converter program to convert such a preliminary service description into an actual execution language script. Fig. 6 illustrates schematically the use of a graphical user interface for composing a service description. In the presented exemplary case the graphical user interface consists of a function library 601 and a workspace 602. The task of the person that composes a service description is to select suitable functions from the function library 601 and to use them to build a graphical representation (a flow diagram) of the service in the workspace 602. The application development system that offers a graphical user interface to the user is then responsible for collecting pieces of precomposed script that correspond to the selected functions and to combine them in a way that corresponds to the service definition structure built by the user. This kind of computer-aided programming is well known as such from the technology of general-purpose application development systems.

A programmatic service definition scenario, regardless of its actual implementation, can rely heavily on preprogrammed library functions. These are passages of execution language script that have been written to both fulfil an actual functional task and to contain an application interface that enables their use as building blocks of any desired service command flow. An example of a library function could be a function "acquire secure connection", which would be built to detect and utilize any currently available possibilities for setting up a secure communication connection to a given destination address. The functional task of such an exemplary library function would be the actual setting up of a connection, while the application interface would consist of a standardized input format in which the function would accept the destination address from any freely selected other part of execution language script, as well as the inter-process communication means by which the function would detect the presence and availability of various means of secure communication. A library function component must typically declare itself to be a "subtask" for an operating system to be able to handle it properly.

A third possible service definition scenario is definition by the service provider. This scenario does not necessarily differ much from a programmatic service definition in actual implementation, but the important difference is that a service provider and not a user is responsible for defining the service and for offering it to be used by users. From the user's point of view this is by far the simplest way of service definition, with the obvious obverse that it does not offer as much possibility of tailoring the services exactly to individual needs. The user may acquire such a service for example by bringing his mobile terminal to a branch office of the service provider for installation, or by downloading the required script(s) from a web page. A service defined by the service provider might well consist of two parts, of which only a simple activating (and potentially result displaying) part runs in the user's mobile terminal and a more complicated processing part runs in the service provider's server.

Figs. 7, 8 and 9 illustrate certain principal alternatives for service execution scenarios. Fig. 7 is a simple terminal-based execution scenario that may include an optional earlier information transfer step 701 from a service provider's server to a mobile terminal, but after that all steps take place locally at the mobile terminal. At step 702 the user activates the use of a service by giving a simple activation command. At step 703 the mobile terminal parses a script that represents the selected service, and at step 704 it processes the service according to the executable instructions that resulted from the parsing step. At step 705 the results are displayed to the user.

Fig. 8 illustrates a more interactive service execution scenario, in which also the server has an active role. At step 801 the user of a mobile terminal activates the use of a service by giving a simple activation command. At step 802 the mobile terminal parses a script that represents the selected service, and at step 803 it formulates a request message to the server according to the executable instructions that resulted from the parsing step. At step 804 the request message is transmitted to the server, which processes the request at step 805 and transmits a response at step 806. At step 807 the mobile terminal receives and processes the response message to put the results of the service into a form in which they can be presented to the user, and at step 808 the results are displayed.

Both service execution scenarios presented in figs. 7 and 8 require the mobile terminal to possess full script parsing capability. However, an alternative approach can be presented where only relatively limited script parsing capability is required from the mobile terminal. In fig. 9 the activation step 901 may be followed by a

script parsing and local processing steps 902 and 903, but in any case the request message that the mobile terminal transmits to the server at step 904 contains also certain passages of yet unparsed script. At step 905 the server parses such passages of script, and/or transforms them into some server specific macro language that is used for controlling the operations of the server. If it is not possible or not desirable to add script parsing capability to the particular server that will provide the actual service, it is possible to arrange step 905 to be performed at an auxiliary "interpreter" device either so that script-containing requests from mobile terminals automatically go first to such an interpreter device or so that when a service provider's server encounters a request that contains script, it forwards it to the interpreter device for parsing the script. The interpreter device then forwards parsed instructions to the actual service-providing server in a form that is acceptable to the latter. An interpreter device may be for example a trusted auxiliary interpreter server hidden behind an actual service provider's server, which has no native parsing capability.

At step 906 the server processes the request, and at step 907 it transmits a response message to the mobile terminal. This response message could comprise passages of script as well. If that is the case, the response processing step 908 at the mobile terminal must include some script parsing as well. At step 909 the results of the service are displayed. If passages of script are transmitted between the mobile terminal and the server, they must be wrapped into some conventional communication protocol: for reasons of compatibility it cannot be required that any other device that takes part in the communication between the mobile terminal and the server should be able to understand or act according to the script.

A division of work between the mobile terminal and the server does not preclude the mobile terminal from performing some service-related processing also during the time when it waits for a response from the server. Using a service at the mobile terminal may comprises two or more parallel processing paths, the execution of which may be concurrent and at least partly independent from each other. All service-related transmissions between the mobile terminal and the server should be encrypted in order to provide security. The communication protocol(s) used for communication over the wireless communication link should include optimization features that take into account the specific nature of wireless links; such optimization features are known and very well developed for example in WAP.

The mechanisms of defining and executing a service according to the invention place little requirements to what is the actual content of the service. However,

certain features of the invention make it especially well suited for financial services. In the following we consider four alternative service usage scenarios that are helpful in understanding certain characteristics that are peculiar to financial services.

5 A first service usage scenario is informational usage, which can be characterised as offering answers to the question "what am I doing?". A user has typically certain financial assets, like bank accounts, a current state of which can be easily characterised by a simple number or character string. However, even such simple information is typically highly confidential, which means that a service for
10 inquiring and providing such information must include security features for ensuring that only an authorized user has access to the information. Typical examples of information of this kind are account balances, remaining monthly credit of a credit card, or the current content of a stocks portfolio.

A second service usage scenario is analytical usage, which is close to the first one described above but is understood to encompass a somewhat wider scope of
15 information. Analytical services may be understood as answering the question "how am I doing?". So instead of or in addition to a remaining monthly credit the user might be interested in accumulated debiting on his accounts over a certain period, classified according to a certain prestored model.

A third service usage scenario is advisory usage, which goes even further from
20 analytical usage in that the user wants to know not only the developments so far, but also certain prospects of the future and/or suggested financial actions. The illustrative question is "what should I do?", and the information looked after in the service may include e.g. investment tips that take into account a predefined risk-taking profile.

25 A fourth service usage scenario is transactional usage, which differs from the three others mentioned above in that instead of or in addition to just receiving information the user wants to actively accomplish something, like paying bills or making investment decisions.

30 Features that are common to all usage scenarios of financial services include a pronounced demand for security, privacy and reliability. The information that is handled is highly confidential and must not be available for unauthorized parties. Transactions that are performed must either succeed completely or not at all, i.e. no transaction should be allowed to be only partly performed. It must be possible to later check and verify a completed transaction. A user must also be able to trust that

the information he gets is correct and really comes from the source from which he expects it to come.

Technology that is used for enhancing security and reliability in electronic communication frequently relies on the use of encryption keys, cryptographically
5 protected authenticity certificates, PINs (Personal Identification Numbers) and other corresponding identifiers that involve complicated character strings that must be either remembered or safely stored. From the technology of secure wireless communication, solutions are known where a mobile communication device includes a so-called electronic safe store, which is a specifically protected memory
10 circuit for safely storing information of this kind. An electronic safe store is even suitable for storing electronic cash, which as a term refers to cryptographically authenticated pieces of digital information that parties in commerce agree to have certain monetary value so that they can be used as tenders for debts. It is possible to give a service according to the invention access to the electronic safe store of the
15 user, which further streamlines the use of financial services in accordance with the invention.

Fig. 10 illustrates a situation where the user first wants to check the balance of his account than then decides to download some electronic cash into his electronic safe store. At step 1001 the user activates a banking service with a simple activation
20 command according to the invention. The user has previously tailored the service so that it always begins with a balance check. The bank will not give the balance information unless the corresponding request message was cryptographically authenticated and contains the correct identifiers; means for cryptographical authentication as well the user's identifiers are stored in the electronic safe store of
25 the mobile terminal. Therefore the terminal requests the appropriate protected information from the safe store at step 1002. The safe store may be further configured so that it will not become available without a PIN number being given online; therefore the optional PIN request and PIN inputting steps 1003 and 1004 are shown in fig. 10. In any case the electronic safe store returns the requested
30 information at step 1005. The other parts of the terminal utilize this information to compose a properly identified and cryptographically authenticated message to the server at step 1006.

Regarding online PIN giving (or similar strictly user-dependent authentication – known alternatives to PIN giving exist and include e.g. electronic fingerprint
35 reading, speech sample analysis and optically scanning the iris of the eye) during the execution of a script, two approaches are possible. The first of them involves

requiring the user to be present and to authenticate himself to the mobile terminal every time when the electronic safe store in the mobile terminal must be accessed. This approach represents ultimate safety, because it vitiates even well-timed attempts of stealing a mobile terminal during the execution of a script, after the
5 legitimate user has authenticated himself but before he has availed himself of any advantage that would involve utilizing the contents of the electronic safe store. The other approach is based on requiring the user to authenticate himself once, after which the script to be currently executed becomes a trusted representative of the user in the sense that it may access the electronic safe store by its own without
10 requiring the user to repeat the authentication step during the period of executing the script. The latter approach is more convenient to the user, and also reasonably safe at least if the time it takes to execute such a "trusted script" remains below a reasonable limit.

Regarding subsequent uses of the same service, two alternatives exist: either a script
15 that has once acquired the status as the user's trusted representative may keep that status ever since (or for a predetermined time, or predetermined number of times of using the service), or then the status as the user's trusted representative only lasts until the end of that particular occasion of using the service, in which latter case the script has to re-acquire the status every time when the script is started anew.

20 It is also possible to utilize the electronic safe store to store the script itself or at least major parts of it. In that case the user must authenticate himself at the very beginning of the execution of a script, and the safely stored script is as safe from unauthorized tampering as anything else stored in the electronic safe store.

Above we assumed that the first request message is a balance check, so after having
25 verified the authenticity of the requesting user the server responds at step 1007. The response message comes in cryptographically protected form, and the terminal is not able to decrypt and verify it without first consulting the electronic safe store again at steps 1008 and 1009. When the balance information is ready in plaintext form, the terminal displays it to the user at step 1010.

30 We assume that one part of the executable instructions that resulted from a previous script parsing step at the mobile terminal was particularly directed to temporarily modify the user interface of the mobile terminal so that at least certain softkeys are taken to the control of the service in question. This means that when the first results are displayed at step 1010, the selections that the user can make are readily
35 available for him by pressing only one key or otherwise activating a very simple

part of the user interface. In other words, in a mobile terminal equipped with softkeys, together with displaying the requested account balance there are displayed texts and/or icons next to the softkeys that indicate, what simple functional alternatives are now available for selection through the press of a softkey.

5 The subject of giving the user the possibility of altering the course of using a service at a particular time deserves some further consideration. It is naturally possible to compose a script the parsing of which produces a relatively complicated program complete with branching points at which a further selection command received from a user will cause the further execution of the program to follow one of a number of
10 alternative routes. However, such an approach could easily result in a situation where a number of similar functionalities could exist parallelly as parts of different scripts. Fig. 11 illustrates schematically an alternative approach. The horizontal bar 1101 illustrates the execution of a parsed script, advancing from the left to the right, so that at moment 1102 the user causes the execution to start by giving a very
15 simple activation command according to the invention. The use of the service in question proceeds as determined in the script until moment 1103, at which the user is given the possibility of making a selection. It is possible to define a default case according to which no selection or the simplest selection possible expressed at moment 1103 causes the use of the service to proceed according to a basic
20 alternative defined in the script itself. Another selection made by the user and expressed in a simple selection command given through the user interface causes the use of the service to divert into a client program 1111, the execution of which starts at moment 1112.

The client program 1111 may be defined in another script, or it may be some
25 completely different, even device-dependent other functionality like a messaging application. In fig. 11 we assume that the client program 1111 in turn offers the user certain selections regarding its termination at alternative moments 1113, 1114 or 1115. Terminating the use of the client program at any of these moments causes a return to the use of the basic service; here we have additionally assumed that
30 terminating the use of the client program at each of these moments causes a return to a different point 1106, 1105 or 1107 respectively of the basic service.

Another exemplary client program is illustrated as the block 1121. It is started if the user expresses a certain selection at a point 1104 of using the basic service that is defined in the script. Executing the simple, no-selections-involved second client
35 program proceeds from moment 1122 to moment 1123, after which a return occurs

to exactly that point in the "master" service at which a diversion to the client program 1121 occurred.

5 The task of a client program may be almost anything. For example, a client program may be responsible for arranging a more elaborate communication between a user and his mobile terminal than what is possible or reasonable to realize by means of a simple script. If a suitable client program is not permanently resident in the mobile terminal of a user, making a certain selection during the execution of a parsed script may trigger the operation of fetching and downloading a suitable client program from a certain predefined location in a network.

10 The execution language script may support a generic interface towards plugin-type external function modules, that may be interchangeable without altering the rest of the steps of using a service. For example, such a generic interface may exist towards plugin-type modules for setting up a safe communication connection. Through the generic interface the execution language script may adapt to the use of any of a
15 multitude of possible ways of setting up a safe communication connection, without paying attention to what are the specific features of the selected method. Modules that could utilize a generic interface definition include but are not limited to communication setup modules, user authentication modules, man-machine-interface maintaining modules and timing modules in the form of clocks, timers and
20 calendars.

The selections at the various selection points may include a feature according to which pressing always the same single key, or otherwise giving always the same simple command that triggered the start of the basic service itself, causes the use of the service to proceed according to the default route, which in fig. 11 means
25 proceeding through the bar 1101 from left to right without making diversions to any client programs. This would be the easiest for a user to learn and understand, since it is natural to associate the simplest and most familiar of commands to the simplest and most often needed way of proceeding through the use of a service.

The results obtained through the use of a client program may have influence on how
30 the use of the basic service proceeds from that moment at which a return to the basic service occurred. For example if a client program was initiated for transmitting a request message to a stock exchange server for getting the latest stock price quote for a certain company and for receiving a response message, and the response message indicated that a certain predefined price limit has been exceeded,
35 the basic service may treat this information as a parameter value that triggers a

suggestion to the user to commence offering his stocks for sale. The use of clients in accordance with the present invention is most typical in services having certain transactional nature.

Referring briefly back to the schematically presented mobile terminal of fig. 2, allowing a service according to the invention to control the user interface means that the reprogrammable user interface command interpreter block 211, is responsive also during the processing of a service. At a certain step of the service processing the processor 201 is arranged to drive a display to show the temporary meanings of the softkeys, so that pressing a particular softkey causes a particular continuation to be selected in processing the service.

In fig. 10 we assume that the user selects downloading electronic cash by pressing the appropriate softkey at step 1011. A further PIN request from the user would probably be unnecessary if only a short time interval has passed since the last one at steps 1003 and 1004, so the task of composing a cryptographically authenticated request message is managed just between the electronic safe store and the rest of the terminal at steps 1012, 1013 and 1014. At step 1015 the server responds by transmitting the requested electronic cash in a message the decryption and authentication of which necessitates a further consultation of the electronic safe store at steps 1016 and 1017. At step 1018 the terminal performs the action ordered in the message from the server, i.e. stores the newly received electronic cash into the electronic safe store. At step 1019 an indication is given to the user about the completion of the requested task. At step 1020 the user terminates the current use of the banking service.

The facts taken into account that a service according to the invention will at least request certain information from the service provider's server, and possibly also cause transactions to be executed, which transactions may involve substantial financial value, it is within the interest of the service provider to make sure that the script(s) that define and control the service are "legal", i.e. acceptable to the service provider. Additionally the user of a mobile terminal certainly wants to ensure that only correct, appropriate and wanted scripts are stored into his terminal. The following considerations apply:

- only signed scripts are acceptable: a mobile terminal will only accept and store a script that contains the digital signature of a recognized service provider;
- the user has the control: the mobile terminal shall ask for the user's approval before storing a script for later use, and the user can at any time check the number

and nature of scripts stored in his terminal and remove possible unwanted or unnecessary scripts;

- the service provider's server shall only respond to communications contacts that have been generated by authenticated, i.e. appropriately signed scripts and that contain some cryptographic verification of this fact.

The service for the use of which a user wants to apply the present invention may well be such that it does not necessarily require its use to be automated in the way described in the present invention. In other words, the user might well use the same service "manually", e.g. by keying in the request messages by hand. It is in the interest of the service provider to be able to discriminate between such occasions where the service is used through a script and such where the use of the service involves manual interaction. A simple way to achieve such discrimination ability is to require that in the course of a user authentication process, a first identifier is selected (e.g. read from an electronic safe store) if manual use is involved, and a second, different identifier is selected instead (or in addition thereto) if a script was responsible for the use of the service.

Fig. 12 illustrates certain overall system aspects concerning the invention. Two communications networks are involved, one of which is a cellular radio network 1201 and the other is a wired data network 1202. In the most advantageous case considering the technology of the priority date of this patent application, a user has two different types of terminals at his disposal: a wireless terminal 1203 for actually using services according to the invention and a more versatile but potentially larger, heavier and clumsier wired terminal 1204. The wireless terminal 1203 contains at least a programmable user interface, which the user may adapt to accept an ultimately simple starting command for the use of a particular service according to the invention, as well as downloading and communication capability for downloading scripts from service definition server(s) and for exchanging messages related to the use of services with service provider's server(s). The wired terminal 1204 contains a network connection and a browser program for contacting service definition servers and for using them to define and edit user-tailored services.

A service definition server 1205 has a network connection to the wired data network 1202 as well as a service definition application that users may access through the use of browser programs. It must also have transmitting capability for transmitting completed service-defining scripts to the mobile terminals of users through the cellular radio network 1201. A service provider's server 1206, which may even be physically the same as a service definition server 1205, must contain the actual

service application as well as a communication interface for bidirectionally communicating with mobile terminals during the course of a user using a certain service.

5 Considering the system framework of fig. 12, in the foregoing we have assumed that the capability of parsing execution language scripts, or at least a part of such capability, resides in the user's mobile terminal 1203. However, it is possible to present an embodiment of the present invention where the mobile terminal can do completely without parsing capability, if enough such capability exists elsewhere in the system. Additionally such an embodiment of the invention assumes that the
10 mobile terminal includes some native mechanism for executing a series of commands in a device-depending command language and for associating the beginning of such executing to some simple user interface command, which the user wants to associate with a certain service. An an example we may assume that the service definition server 1205 is capable of parsing an execution language script
15 into a device-dependent command series. After the process of composing an execution language script has been completed in the service definition server 1205, one still needs to indicate the type of the mobile terminal 1203 to the service definition server 1205. When the latter recognizes the terminal type in question, it may continue by parsing the completed script, so that a subsequent downloading
20 step involves downloading the parsed device-dependent command series instead of the execution language script from the service definition server 1205 to the mobile terminal 1203.

Another alternative is that users may acquire conversion programs to their wired terminals. In an example of a scheme based on such an assumption, when a user has
25 completed the action of defining his personalized way of using a service at the service definition server 1205, and the corresponding script has been composed and duly authenticated, the user may download the script into his wired terminal 1204, convert it into a device-dependent command series and forward the result to his mobile terminal 1203 through the networks 1201 and 1202 or through a local short-
30 distance link. If the steps of defining the service and composing the script took place at the user's wired terminal 1204, no script downloading is needed but the converting and forwarding steps are performed in a similar manner.